Bulletin of Mathematics
Vol. 03, No. 01 (2011), pp. 1–16.

KOLMOGOROV COMPLEXITY: CLUSTERING OBJECTS AND SIMILARITY

Mahyuddin K. M. Nasution

Abstract. *The clustering objects has become one of themes in many studies, and do not few researchers use the similarity to cluster the instances automatically. However, few research consider using Kommogorov Complexity to get information about objects from documents, such as Web pages, where the rich information from an approach proved to be difficult to. In this paper, we proposed a similarity measure from Kolmogorov Complexity, and we demonstrate the possibility of exploiting features from Web based on hit counts for objects of Indonesia Intellectual.*

1. INTRODUCTION

In mathematics, the object is an abstract arising in mathematics, generally is known as *mathematical object*. Commonly they include numbers, permutations, partitions, matrices, sets, functions, and relations. In computer science, these objects can be viewed as binary strings, or strings in forms are words, sentences or documents. Thus we will refer to objects and string interchangeably in this paper. Therefore, sometimes some research also will refer to data as objects or objects as data.

A binary string has the length of the shortest program which can output the string on a universal Turing machine and then stop [1]. A universal Turing machine is an idealized computing device capable of reading, writing,

Received 11-10-2010, Accepted 15-11-2010.
2000 Mathematics Subject Classification: 03D15, 68Q15
Key words and Phrases: Kolmogorov complexity, distance, similarity, singleton, doubleton.

processing instructions and halting [2, 3]. The concept of Turing machine is widely used in theoretical computer science, as computational model based on mathematics to approach some problems of real-world. One of problems is about word sense, mainly about context. This problem appears in some applications like machine translation and text summarization, where mostly the existing system needs to understand the correct meaning (semantics relation) and function of words in natural language. This means that the aquacition of knowledge needs a model to abstracts an incomplete information. Therefore, this paper is to address a tool of measurement based on Kolmogorov complexity for finding relations among objects. We first review, in Section 2, the basic terminologies and the concepts. We state, in Section 3, the fundamental results and we discussion property of similarity in Lemma and Theorem. In Section 4, we study a set of objects from Indonesia intellectuals.

2. RELATED WORK

In mathematics, it is more important that objects be definable in some uniform way, for example as sets. Regardless of actual practice, in order to lay bare the essence of its paradoxes, which has traditionally accorded the management of paradox higher priority to objects, and this needs the faithful reflection of the details of mathematical practice as a justification for defining objects. Turing showed this problem in his famous work on the halting problem that it is impossible to write a computer program which is able to predict if some other program will halt [4, 5]. Thus it is impossible to compute the complexity of a binary string. However there have been methods developed to approximate it, and Kolmogorov complexity is of length of the shortest program which can output the string, where objects can be given literally as like as the human can be represented in DNA [6].

Kolmogorov complexity, also known as algorithm entropy, stochastic complexity, descriptive complexity, Kolmogorov-Chaitin complexity and program-size complexity, is used to describe the complexity or degree of randomness of a binary string. It was independently developed by Andrey N. Kolmogorov, Ray Solomonoff and Gregory Chaitin in the late 1960's [7, 5]. For an introduction and details see the textbook [8].

Definition 2.1 *The* Kolmogorov complexity *of a string x, denoted as $K(x)$, is the length, in bits, of the shortest computer program of the fixed reference computing systems that produces x as output.*

The choice of computing system changes the value of $K(x)$ by at most an additive fixed constant. Since $K(x) \xrightarrow{x} \infty$, this additive fixed constant is an ignorable quantity if x is large. One way to think about the Kolmogorov complexity $K(x)$ is to view it as the length (bits) of the ultimate compressed version from which x can be recovered by a general decompression program. The associated compression algorithm transform x_z back into x or a string very close to x. A lossless compression algorithm is one in which the decompression algorithm exactly computes x from x_z and a lossy compression algorithm is one which x can be approximated given x_z. Usually, the length $|x_z| < |x|$. Using a better compressor results in x_b with no redundant information, usually $|x_b| < |x_z|$, etc. So, lossless compression algorithms are used when there can be no loss of data between compression and decompressin. When $K(x)$ is approximation corresponds to an upper-bound of $K(x)$ [9]. Let C be any compression algorithm and let $C(x)$ be the results of compressing x using C.

Definition 2.2 *The approximate Kolmogorov complexity of x, using C as a compression algorithm, denoted $K_C(x)$, is*

$$K_C(x) = \frac{\text{Length}(C(x))}{\text{Length}(x)} + q = \frac{|C(x)|}{|x|} + q$$

where q is the length in bits of the program which implements C.

If C was able to compress x a great deal then $K_C(x)$ is low and thus x has low complexity. Using this approximation, the similarity between two finite objects can be compared [10, 9].

Definition 2.3 *The information shared between two string x and y, denoted $I(x : y)$, is $I(x : y) = K(y) - K(y|x)$, where $K(y|x)$ is Kolmogorov complexity of y relative to x, is the length of the shortest program which can output y if $K(x)$ is given as additional input to the program.*

Previous classification research using Kolmogorov complexity has been based on the similarity metric developed [11, 12]. Two strings which are similar share patterns and can be compressed more when concatenated than separately. In this way the similarities between data can be measured. This method has been successfully used to classify documets, music, email, and those are of: network traffic, detecting plagiarism, computing similarities between genomes and tracking the evaluation of chain letters [13, 14, 15, 16, 17, 18].

Table 1: Data compression

w	Key	$C(w)$	$\|C(w)\|$	$\|w\|$	$K_P(w)$
s_1	$k_1 = 0100$	$k_1k_2k_1k_3k_1k_4k_5k_6k_5k_5+$	34	40	0.85
	$k_2 = 1101$	"$k_1 = 0100$ $k_2 = 1101$ $k_3 = 0001$			
	$k_3 = 0001$	$k_4 = 1000$ $k_5 = 0101$ $k_6 = 1010$"			
	$k_4 = 1000$				
	$k_5 = 0101$				
	$k_6 = 1010$				
s_2	$k_1 = 0100$	$k_1k_1k_1k_1k_1k_7k_1k_8+$	20	32	0.625
	$k_7 = 1001$	"$k_1 = 0100$ $k_7 = 1001$ $k_8 = 1110$"			
	$k_8 = 1110$				
s_3	$k_5 = 1001$	$k_5k_6k_5k_5k_1k_1k_1k_7+$	24	32	0.75
	$k_6 = 1010$	"$k_5 = 1001$ $k_6 = 1010$ $k_1 = 0100$			
	$k_1 = 0100$	$k_7 = 1001$"			
	$k_7 = 1001$				
$s_1\|s_2$	$k_2 = 1101$	$k_1k_2k_1k_3k_1k_4k_5k_6k_5k_5+$	30	40	0.75
	$k_3 = 0001$	"$k_2 = 1101$ $k_3 = 0001$ $k_4 = 1000$			
	$k_4 = 1000$	$k_5 = 0101$ $k_6 = 1010$"			
	$k_5 = 0101$				
	$k_6 = 1010$				
$s_1\|s_3$	$k_2 = 1101$	$k_1k_2k_1k_3k_1k_4k_5k_6k_5k_5+$	22	40	0.55
	$k_3 = 0001$	"$k_2 = 1101$ $k_3 = 0001$ $k_4 = 1000$"			
	$k_4 = 1000$				

3. DISTANCE, METRIC AND SIMILARITY

Suppose there is a pattern matching algorithm based on compressing each consecutive set of four binary digits (hexadecimal). Let C is the program that performs this compression. For each string w, C generates a key of single characters which corresponding to sets of four digits.

Let $s_1 =$ "$b_0b_1b_1b_0b_1b_1b_1b_0$" will generate keys $k_1 = b_0b_1b_1b_0$ and $k_2 = b_1b_1b_1b_0$. The compressed string is composed of the representation plus the key, i.e. $k_1k_2 +$ "$k_1 = b_0b_1b_0b_4\ k_2 = b_1b_1b_1b_0$". Suppose a second string $s_2 = b_0b_1b_1b_0b_1b_1b_0b_0$ and keys are $k_1 = b_0b_1b_1b_0$ and $k_3 = b_1b_1b_0b_0$, and then the compressed string of s_2 is $k_1k_3 +$ "$k_1 = b_0b_1b_0b_4\ k_3 = b_1b_1b_0b_0$. We can write $C(s_1|s_2) = k_1k_2 +$ "$k_2 = b_1b_1b_1b_0$". Thus $|C(s_1|s_2)| < |C(s_1)|$ because there is a similar pattern in s_1 and s_2. For example, we have three strings

$$\begin{aligned} s_1 &= 0100\ 1101\ 0100\ 0001\ 0100\ 1000\ 0101\ 1010\ 0101\ 0101, \\ s_2 &= 0100\ 0100\ 0100\ 0100\ 0100\ 1001\ 0100\ 1110, \text{ and} \\ s_3 &= 1001\ 1010\ 1001\ 1001\ 0100\ 0100\ 0100\ 1001. \end{aligned}$$

We can compress each string individually and also the results of compressing s_1 using the keys already developed for s_2 and s_3, Table 1.

$$\begin{aligned} I_C(s_2 : s_1) &= K_P(s_1) - K_P(s_1|s_2) = 0.85 - 0.75 = 0.10 \\ I_C(s_3 : s_1) &= K_P(s_1) - K_P(s_1|s_3) = 0.85 - 0.55 = 0.30 \end{aligned}$$

Thus $I_C(s_3 : s_1) > I_C(s_2 : s_1)$ is that s_1 and s_3 share more information than s_1 and s_2. This defines that the information shared between two strings can be approximated by using a compression algorithm C. Therefore, the length of the shortest binary program in the reference universal computing system such that the program computes output y from input x, and also ouput x from input y, called *information distance* [19, 11, 12].

Definition 3.4 *Let X be a set. A function $E : X \times X \to \mathbf{R}$ is called* information distance *(or dissimilarity) on X, denoted $E(x, y)$, i.e. $E(x, y) = K(x|y) - \min\{K(x), K(y)\}$ for all $x, y \in X$, it holds:*

1. *$E(x, y) \geq 0$, (non-negativity);*
2. *$E(x, y) = E(y, x)$, (symmmetry) and;*
3. *$E(x, y) \leq E(x, z) + E(z, y)$, (transitivity).*

This distance $E(x, y)$ is actually a metric, but on properties of information distance these distances that are nonnegative and symmetric, i.e. for considering a large class of admissible distances, whereas computable in the sense that for every such distance J there is a prefix program that has binary length equal to the distance $D(x, y)$ between x and y. This means that

$$E(x, y) \leq D(x, y) + c_D$$

where c_D is a constant that depends only on D but not on x and y. Therefore, there are some distances related to one another with features that because it is not suitable. Thus we need to normalize the information distance.

Definition 3.5 *Normalized information distance, denoted $N(x|y)$, is*

$$N(x|y) = \frac{K(x|y) - \min\{K(x), K(y)\}}{\max\{K(x), K(y)\}}$$

such that $N(x|y) \in [0, 1]$.

Analogously, if C is a compressor and we use $C(x)$ to denote the length of the compressed version of a string x, we define *normalized compression distance.*

Definition 3.6 *Normalized compression distance, denoted $N_c(x|y)$, is*

$$N_c(x|y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

where for convenience the pair $(x|y)$ is replaced by the concatenation xy.

From Table 1, we calculate $N_c(s_1|s_2) = \frac{30-20}{34} = 0.294118$, whereas $N_c(s_1|s_3) = \frac{22-24}{34} = -0.058824$.

The string give a name to object, like "the three-letter genome of 'love'" or "the text of *The Da Vinci Code* by Dan Brown", also there are objects that do not have name literally, but acquire their meaning from their contexts in background common knowledge in humankind, like "car" or "green". The objects are classified by word, the words as objects are classified in the sentences where it represented how the society used the objects, and the words and the sentences are classified in documents.

Definition 3.7 *$W = \{w_1, \ldots, w_v\}$ represents the number of unique words (i.e., vocabulary) and a word as grain of vocabulary indexed by $\{1, \ldots, v\}$.*

Definition 3.8 *A document d is a sequence of n words denoted by* $\mathbf{w} = \{w_i | i = 1, \ldots, n\}$*, where w_n denotes the nth word in a document.*

Definition 3.9 *A corpus is a collection of m documents denoted by* $\mathbf{D} = \{d_j | j = 1, \ldots, m\}$*, where d_m denotes the mth document in a corpus.*

In real world, the corpus is divided two kind: annotated corpus and large corpus. The last definition is a representation of body of informatin physically limited by designing capacity for managing documents. Unfortunately, the modelling collection of document as the annotated corpus not only need more times and much cost to construct and then to manage it, but also this modelling eliminate dynamic property from it. Other side, the collection of digital documents on Internet as web have been increased extremely and changed continuously, and to access them generally based on indexes.

Let the set of document indexed by system tool be Ω, where its cardinality is $|\Omega|$. In our example, $\Omega = \{k_1, \ldots, k_8\}$, and $|\Omega| = 13$. Let every term x defines *singleton event* $\mathbf{x} \subseteq \Omega$ of documents that contain an occurence of x. Let $P : \Omega \to [0, 1]$ be the uniform mass probability function. The probability of event $\mathbf{x}$ is $P(\mathbf{x}) = |\mathbf{x}|/|\Omega|$. Similarly, for terms x AND y, the *doubleton event* $\mathbf{x} \cap \mathbf{y} \subseteq \Omega$ is the set of documents that contain both term x and term y (co-occurrence), where their probability together is $P(\mathbf{x} \cap \mathbf{y}) = |\mathbf{x} \cap \mathbf{y}|/|\Omega|$. Then, based on other Boolean operations and rules can be developed their probability of events via above singleton or doubleton. From Table 1, we know that term k_1 has $|k_1| = 3$ in s_1, $|k_1| = 6$ in s_2 and $|k_3| = 3$ in s_3. Probability of event k_1 is $P(k_1) = 3/13 = 0.230769$ because term k_1 is occurence in three string as document. Probability of event $\{k_1, k_5\}$ is $P(\{k_1, k_5\}) = 2/13 = 0.153846$ from s_1 dan s_3.

It has been known that the strings x where the complexity $C(x)$ represents the length of the compressed version of x using compressor C, for a search term x, search engine code of length $S(x)$ represents the shortest expected prefix-code word length of the associated search engine event $\mathbf{x}$. Therefore, we can rewrite the equation on Definition 3.6 as

$$N_S(x, y) = \frac{S(x|y) - \min\{S(x), S(y)\}}{\max\{S(x), S(y)\}},$$

called *normalized search engine distance*.

Let a probability mass function over set $\{\{x, y\} : x, y \in \mathcal{S}\}$ of searching terms by search engine based on probability events, where $\mathcal{S}$ is universal of

singleton term. There are $|\mathcal{S}|$ singleton terms, and 2-combination of $|\mathcal{S}|$ doubleton consisting of a pair of non-identical terms, $x \neq y$, $\{x, y\} \subseteq \mathcal{S}$. Let $z \in \mathbf{x} \cap \mathbf{y}$, if $\mathbf{x} = \mathbf{x} \cap \mathbf{x}$ and $\mathbf{y} = \mathbf{y} \cap \mathbf{y}$, then $z \in \mathbf{x} \cap \mathbf{x}$ and $z \in \mathbf{y} \cap \mathbf{y}$. For $\Psi = \sum_{\{x,y\} \subseteq \mathcal{S}} |\mathbf{x} \cap \mathbf{y}|$, it means that $|\Psi| \geq |\Omega|$, or $|\Psi| \leq \alpha|\Omega|$, α is constant of search terms. Consequently, we can define $p(x) = \frac{P(\mathbf{x})|\Omega|}{|\Psi|} = \frac{|\mathbf{x}|}{|\Psi|}$, and for $\mathbf{x} = \mathbf{x} \cap \mathbf{x}$, we have $p(x) = \frac{P(\mathbf{x})|\Omega|}{|\Psi|} = \frac{P(\mathbf{x} \cap \mathbf{x})|\Omega|}{|\Psi|} = p(x, x)$ atau $p(x, x) = \frac{|\mathbf{x} \cap \mathbf{x}|}{|\Psi|}$.

For $P(\mathbf{x}|\mathbf{y})$ means a conditional probability, so $p(x) = p(x|x)$ and $p(x|y) = P(\mathbf{x} \cap \mathbf{y})|\Omega|/|\Psi|$. Let $\{k_1, k_5\}$ is a set, there are three subsets contain k_1 or k_5: $\{k_1\}$, $\{k_5\}$, and $\{k_1, k_5\}$. Let we define an analogy, where $S(x)$ and $S(x|y)$ mean $p(x)$ and $p(x|y)$. Based on normalized search engine distance equation, we have

$$\begin{aligned} N_S(x, y) &= \frac{|\mathbf{x} \cap \mathbf{y}|/|\Psi| - \min(|\mathbf{x}|/|\Psi|, |\mathbf{y}|/|\Psi|)}{\max(|\mathbf{x}|/|\Psi|, |\mathbf{y}|/|\Psi|)} \\ &= \frac{|\mathbf{x} \cap \mathbf{y}| - \min(|\mathbf{x}|, |\mathbf{y}|)}{\max(|\mathbf{x}|, |\mathbf{y}|)} \end{aligned} \tag{1}$$

Definition 3.10 *Let X be a set. A function $s : X \times X \rightarrow \mathbf{R}$ is called* similarity *(or proximity) on X if s is non-negative, symmetric, and if $s(x, y) \leq s(x, x)$, $\forall x, y \in X$, with an equality if and only if $x = y$.*

Lemma 3.1 *If $x, y \in X$, $s(x, y) = 0$ is a minimum weakest value between x and y and $s(x, y) = 1$ is a maximum strongest value betweem x and y, then a function $s : X \times X \rightarrow [0, 1]$, such that $\forall x, y \in X$, $s(x, y) \in [0, 1]$.*

Proof 3.1 *Let $|X|$ is a cardinality of X, and $|x|$ is a number of x occured in X, the ratio between X and x is $0 \leq |\mathbf{x}|/|\mathbf{X}| \leq 1$, where $|\mathbf{x}| \leq |\mathbf{X}|$.*

The $s(x, x)$ means that a number of x is compared with x-self, i.e. $|\mathbf{x}|/|\mathbf{x}| = 1$, or $\forall x \in X$, $|\mathbf{X}|/|\mathbf{X}| = 1$. Thus $1 \in [0, 1]$ is a closest value of $s(x, x)$ or called a maximum strongest value.

In other word, let $z \notin X$, $|\mathbf{z}| = 0$ means that a number of z do not occur in X, and the ratio between z and X is 0, i.e., $|\mathbf{z}|/|\mathbf{X}| = 0$. Thus $0 \in [0, 1]$ is a unclosest value of $s(x, z)$ or called a minimum weakest value.

The $s(x, y)$ means that a ratio between a number of x occured in X and a number of y occured in X, i.e., $|\mathbf{x}|/|\mathbf{X}|$ and $|\mathbf{y}|/|\mathbf{X}|$, $x, y \in X$. If $|\mathbf{X}| = |\mathbf{x}| + |\mathbf{y}|$, then $|\mathbf{x}| < |\mathbf{X}|$ and $|\mathbf{y}| < |\mathbf{X}|$, or $(|\mathbf{x}|/|\mathbf{X}|)(|\mathbf{y}|/|\mathbf{X}|) = |\mathbf{x}||\mathbf{y}|/|\mathbf{X}|^2 \leq 1$ and $|\mathbf{x}||\mathbf{y}|/|\mathbf{X}|^2 \geq 0$. Thus $s(x, y) \in [0, 1]$, $\forall x, y \in X$.

Theorem 3.1 *$\forall x, y \in X$, the similarity of x and y in X is*

$$s(x, y) = \frac{2|\mathbf{x} \cap \mathbf{y}|}{|\mathbf{x}| + |\mathbf{y}|} + c$$

where c is a constant.

Proof 3.2 *By Definition 3.4 and Definition 3.10, the main transforms is used to obtain a distance (dissimilarity) d from a similarity s are* $d = 1 - s$, *and from (1) we obtain* $1 - s = \frac{|\mathbf{x} \cap \mathbf{y}| - \min(|\mathbf{x}|,|\mathbf{y}|)}{\max(|\mathbf{x}|,|\mathbf{y}|)}$.

Based on Lemma 3.1, for maximum value of s is 1, we have $0 = \frac{|\mathbf{x} \cap \mathbf{y}| - \min(|\mathbf{x}|,|\mathbf{y}|)}{\max(|\mathbf{x}|,|\mathbf{y}|)}$ *or* $|\mathbf{x} \cap \mathbf{y}| = \min(|\mathbf{x}|, |\mathbf{y}|)$. *For minimum value of s is 0, we obtain*

$$1 = \frac{|\mathbf{x} \cap \mathbf{y}| - \min(|\mathbf{x}|, |\mathbf{y}|)}{\max(|\mathbf{x}|, |\mathbf{y}|)}$$

or

$$\begin{aligned} |\mathbf{x} \cap \mathbf{y}| &= \max(|\mathbf{x}|, |\mathbf{y}|) + \min(|\mathbf{x}|, |\mathbf{y}|) \\ &= |\mathbf{x}| + |\mathbf{y}| \end{aligned}$$

or $1 = (|\mathbf{x} \cap \mathbf{y}|)/(|\mathbf{x}| + |\mathbf{y}|)$. *We know that* $|\mathbf{x}| + |\mathbf{y}| > |\mathbf{x} \cap \mathbf{y}|$, *because their ratios are not 1. If* $x = y$, *then* $|\mathbf{x} \cap \mathbf{y}| = |\mathbf{x}| = |\mathbf{y}|$, *its consequence is* $1 = (2|\mathbf{x} \cap \mathbf{y}|)/(|\mathbf{x}| + |\mathbf{y}|)$. *Therefore, we have* $s = \frac{2|\mathbf{x} \cap \mathbf{y}|}{|\mathbf{x}| + |\mathbf{y}|} + 1$, *and* $c = 1$, *or*

$$s = \frac{2|\mathbf{x} \cap \mathbf{y}|}{|\mathbf{x}| + |\mathbf{y}|} + c.$$

For normalization, we define $|\mathbf{x}| = \log f(x)$ and $2|\mathbf{x} \cap \mathbf{y}| = \log(2f(x, y))$, and the similarity on Definition 3.11 satisfies Theorem 3.1.

Definition 3.11 *Let similarity metric I is a function* $s(x, y) : X \times X \to [0, 1]$, $x, y \in X$. *We define similarity metric M as follow:*

$$s(x, y) = \frac{\log(2f(x, y))}{\log(f(x) + f(y))}$$

In [12], they developed Google similarity distance for Google search engine results based on Kolmogorov complexity:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}}$$

For example, at the time, a Google search for "horse", returned 46,700,000 hits, for "rider" was returned 12,200,000 hits, and searching for the pages where both "rider" and "rider" occur gave 2,630,000. Google indexed $N = 8,058,044,651$ web pages, and $NGD(horse, rider) \approx 0.443$. Using equation in Defenition 10, we have $(s, y) \approx 0.865$, about two times the results of Google similarity distance. At the time of doing the experment, we have

Table 2: Similarity for two results.

Search engine	x (= "horse")	y (= "rider")	x AND y	$s(x,y)$
Google	150,000,000	57,000,000	12,400,000	0.889187
Yahoo!	737,000,000	256,000,000	52,000,000	0.891084

150,000,000 and 57,000,000 for "horse" and "rider" from Google, respectively. While the number of hits for the search both terms "horse" AND "rider" is 12,400,000, but we will not have N exactly, aside from predicting it. We use similarity metric M for comparing returned results of Google and Yahoo!, Table 2.

4. APPLICATION AND EXPERIMENT

Given a set of objects as points, in this case a set of authors of Indonesian Intellectuals from *Commissie voor de Volkslectuur* and their works (Table 3), and a set of authors of Indonesian Intellectuals from New Writer with their works (Tabel 4).

The authors of *Commissie voor de Volkslectuur* are a list of 9 person names:

{(1) Merari Siregar; (2) Marah Roesli; (3) Muhammad Yamin; (4) Nur Sutan Iskandar; (5) Tulis Sutan Sati; (6) Djamaluddin Adinegoro; (7) Abas Soetan Pamoentjak; (8) Abdul Muis; (9) Aman Datuk Madjoindo}.

While the authors of New Writer are 12 peoples, i.e.,

{(i) Sutan Takdir Alisjahbana; (ii) Hamka; (iii) Armijn Pane; (iv) Sanusi Pane; (v) Tengku Amir Hamzah; (vi) Roestam Effendi; (vii) Sariamin Ismail; (viii) Anak Agung Pandji Tisna; (ix) J. E. Tatengkeng; (x) Fatimah Hasan Delais; (xi) Said Daeng Muntu; (xii) Karim Halim}.

In a space provided with a distance measure, we extract more information from Web using Yahoo! search engines, then we build the associated distance matrix which has entries the pairwise distance between the objects laying on Definition 3.11. We define some type of relations between author and his/her works in 9 categories: (1) unclose (value < 0.11), (2) weakest ($0.11 \leq$ value < 0.22), (3) weaker ($0.22 \leq$ value < 0.33), (4) weak ($0.33 \leq$

Table 3: Indonesian Intellectual of *Commissie voor de Volkslectuur*

id	Name of Indesian Intellectual	Year	Author	Value	Type
a.	Azab dan Sengsara	1920	1	0.7348	7
b.	Binasa kerna Gadis Priangan	1931	1	0.6569	6
c.	Cinta dan Hawa Nafsu		1	0.4357	4
d.	Siti Nurbaya	1922	2	0.5706	6
e.	La Hami	1924	2	0.3831	4
f.	Anak dan Kemenakan	1956	2	0.5461	5
g.	Tanah Air	1922	3	0.6758	7
h.	Indonesia, Tumpah Darahku	1928	3	0.5183	5
i.	Kalau Dewi Tara Sudah Berkata		3	0.4582	5
j.	Ken arok dan Ken Dedes	1934	3	0.4922	5
k.	Apa Dayaku karena Aku Seorang Perempuan	1923	4	0.5374	5
l.	Cinta yang Membawa Maut	1926	4	0.8189	8
m.	Salah Pilih	1928	4	0.7476	7
n.	Karena Mentua	1932	4	0.6110	6
o.	Tuba Dibalas dengan Susu	1933	4	0.5918	6
p.	Hulubalang Raja	1934	4	0.7759	7
q.	Katak Hendak Menjadi Lembu	1935	4	0.8424	8
r.	Tak Disangka	1923	5	0.4811	5
s.	Sengsara Membawa Nikmat	1928	5	0.6006	6
t.	Tak Membalas Guna	1932	5	0.5139	5
u.	Memutuskan Pertalian	1932	5	0.6150	6
v.	Darah Muda	1927	6	0.3632	4
w.	Asmara Jaya	1928	6	0.3896	4
x.	Pertemuan	1927	7	0.2805	2
y.	Salah Asuhan	1928	8	0.7425	7
z.	Pertemuan Djodoh	1933	8	0.4376	4
aa.	Menebus Dosa	1932	9	0.4531	5
ab.	Si Cebol Rindukan Bulan	1934	9	0.7516	7
ac.	Sampaikan Salamku Kepadanya	1935	9	0.5786	6

Table 4: Indonesian Intellectual of New Writer

id	Name of Indoensian Intelectual	Year	Author	Value	Type
A.	Dian Tak Kunjung Padam	1932	i	0.6372	6
B.	Tebaran Mega (kumpulan sajak)	1935	i	0.6189	6
C.	Layar Terkembang	1936	i	0.7494	7
D.	Anak Perawan di Sarang Penyamun	1940	i	0.6095	6
E.	Di Bawah Lindungan Ka'bah	1938	ii	0.4302	4
F.	Tenggelamnya Kapal van der Wijck	1939	ii	0.7245	7
G.	Tuan Direktur	1950	ii	0.6506	6
H.	Didalam Lembah Kehidoepan	1940	ii	0.3723	4
I.	Belenggu	1940	iii	0.6007	6
J.	Jiwa Berjiwa		iii	0.4669	5
K.	Gamelan Djiwa (kumpulan sajak)	1960	iii	0.6055	6
L.	Djinak-djinak Merpati (sandiwara)	1950	iii	0.6378	6
M.	Kisah Antara Manusia (kumpulan cerpen)	1953	iii	0.5380	5
O.	Pancaran Cinta	1926	iv	0.5393	5
P.	Puspa Mega	1927	iv	0.5681	6
Q.	Madah Kelana	1931	iv	0.6477	6
R.	Sandhyakala Ning Majapahit	1933	iv	0.6035	6
S.	Kertajaya	1932	iv	0.4872	5
T.	Nyanyian Sunyi	1937	v	0.5249	5
U.	Begawat Gita	1933	v	0.3175	2
V.	Setanggi Timur	1939	v	0.5058	5
W.	Bebasari: toneel dalam 3 pertundjukan		vi	0.5918	6
X.	Pertjikan Permenungan		vi	0.4988	5
Y.	Kalau Tak Untung	1933	vii	0.3611	4
Z.	Pengaruh Keadaan	1937	vii	0.3655	4
AA.	Ni Rawit Ceti Penjual Orang	1935	viii	0.7906	8
AB.	Sukreni Gadis Bali	1936	viii	0.7492	7
AC.	I Swasta Setahun di Bedahulu	1938	viii	0.7882	8
AD.	Rindoe Dendam	1934	ix	0.6034	6
AE.	Kehilangan Mestika	1935	x	0.5132	5
AF.	Karena Kerendahan Boedi	1941	xi	0.8084	8
AG.	Pembalasan		xi	0.4057	4
AH.	Palawija	1944	xii	0.3886	4

	1	2	3	4	5	6	7	8	9	i	ii	iii	iv	v	vi	vii	viii	ix	x	xi	xii	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac
1		6	5	6	7	5	5	5	7	7	5	6	6	5	6	6	6	6	5	6	4	7	6	4	6	4	6	4	4	5	6	5	4	4	6	4	5	5	4	6	5	5	4	4	5	5	6	5	6	6
2	6		5	6	6	6	5	5	6	6	5	6	6	5	6	5	6	6	5	6	4	5	6	4	6	4	5	5	5	5	6	5	4	5	5	4	5	5	5	5	5	5	5	5	5	5	6	5	5	5
3	5	5		8	5	5	2	8	4	6	6	5	6	7	5	4	4	5	4	4	7	5	4	6	6	5	5	7	5	5	5	4	6	6	4	4	5	5	6	6	6	5	7	6	7	6	4	5	4	4
4	6	6	8		7	5	4	9	6	8	7	6	7	8	6	5	5	6	5	5	7	7	5	7	7	5	6	7	6	7	6	5	8	7	6	6	7	8	7	7	8	7	7	7	8	7	5	6	5	6
5	7	6	5	7		6	5	6	7	6	5	6	6	5	6	7	7	6	5	7	4	6	6	5	5	4	6	5	5	5	6	5	5	5	6	5	6	5	5	6	5	6	5	5	5	5	7	5	6	6
6	5	6	5	5	6		5	5	6	6	4	5	5	4	6	7	5	5	5	6	4	4	6	2	4	2	5	4	5	5	5	4	4	2	5	4	4	4	4	4	4	4	4	4	4	4	6	4	5	5
7	5	5	2	4	5	5		4	7	5	2	5	5	2	5	7	6	5	8	8	2	4	8	2	4	2	4	2	4	4	5	4	2	2	5	4	4	4	2	4	2	4	2	2	2	2	7	4	8	4
8	5	5	8	9	6	5	4		5	7	7	6	7	8	5	5	5	5	4	5	8	7	4	8	7	7	6	8	5	6	5	5	8	7	5	5	6	7	8	7	8	6	8	8	8	7	4	7	4	6
9	7	6	4	6	7	6	7	5		6	4	6	6	4	6	7	7	6	7	8	2	7	5	8	4	5	4	5	4	4	5	6	5	4	2	6	4	5	5	4	5	4	4	4	4	4	8	5	7	6
i	7	6	6	8	6	6	5	7	6		6	8	8	5	6	6	6	6	5	6	4	6	5	5	6	4	6	5	5	5	6	5	5	5	5	5	5	5	5	5	5	6	5	5	6	6	5	5	5	5
ii	5	5	6	7	5	4	2	7	4	6		5	5	6	4	4	4	4	2	4	7	6	4	7	6	5	6	7	4	5	4	4	7	7	5	4	6	6	6	6	7	5	7	6	8	6	4	6	2	5
iii	6	6	5	6	6	5	5	6	6	8	5		8	5	6	6	6	6	5	6	4	6	6	5	5	4	5	5	5	5	6	5	5	4	5	5	5	5	4	5	5	5	5	5	5	5	6	5	5	5
iv	6	6	6	7	6	5	5	7	6	8	5	8		5	6	5	6	6	5	6	5	5	5	5	5	4	5	5	5	5	6	5	5	5	5	5	5	5	5	5	5	5	5	5	6	5	5	5	5	5
v	5	5	7	8	5	4	2	8	4	5	6	5	5		5	4	5	5	4	5	7	5	4	6	5	4	5	6	4	5	5	4	6	6	5	4	6	6	6	5	6	5	6	6	7	6	4	5	4	5
vi	6	6	5	6	6	6	5	5	6	6	4	6	6	5		6	6	6	5	6	4	5	6	4	5	4	5	4	5	5	6	5	4	4	5	4	5	4	4	5	4	5	4	4	4	4	6	4	5	5
vii	6	5	4	5	7	7	7	5	7	6	4	6	5	4	6		6	6	7	7	4	4	7	4	4	2	5	2	5	5	5	5	4	2	5	5	5	4	4	4	4	5	4	4	4	4	7	4	7	5
viii	6	6	4	5	7	5	6	5	7	6	4	6	6	5	6	6		6	6	7	4	5	7	4	5	4	5	4	4	5	6	5	2	2	6	4	5	5	4	5	4	5	4	4	4	4	7	4	6	5
ix	6	6	5	6	6	5	5	5	6	6	4	6	6	5	6	6	6		5	6	2	5	6	4	5	4	5	4	5	5	6	5	4	4	5	5	5	5	4	5	4	5	4	4	4	4	6	4	5	5
x	5	5	4	5	5	5	8	4	7	5	2	5	5	4	5	7	6	6		7	2	4	7	4	4	2	4	2	4	4	5	4	4	2	5	4	4	4	2	4	2	4	2	2	2	4	7	4	7	4
xi	6	6	4	5	7	6	8	5	8	6	4	6	6	5	6	7	7	6	7		2	5	9	4	5	4	5	2	4	5	6	5	2	2	6	4	5	5	4	5	4	5	4	4	4	4	8	4	7	6
xii	4	4	7	7	4	4	2	8	2	4	7	4	5	7	4	4	4	6	2	2		5	2	6	5	6	4	7	4	4	2	4	6	6	4	4	5	5	6	6	6	5	7	7	7	6	2	5	2	4

Fig. 1: Matrix of relations.

value < 0.44), (5) midle (0.44 ≤ value < 0.56), (6) strong (0.56 ≤ value < 0.67), (7) stronger (0.67 ≤ value < 0.78), (8) strongest (0.78 ≤ value

< 0.89), and (9) close (value ≥ 0.89).

Specifically, some of Indonesia intellectuals of *Commissie voor de Volkslectuur* and New Writer be well-known works, mainly works from famous authors whose popularity in society, but also there are visible works because familiar name (or same name), for example the story of "Begawat Gita" from Tengku Amir Hamzah, or because the given name frequently appear as words in other people work or web pages, for example the story of "Pertemuan" from Abas Soetan Pamoentjak, see Table 3 and Table 4.

Generally, the appearance of strong interactions in web pages among *Commissie voor de Volkslectuur* and New Writer. This situation derive from the time the works appear in the same range of years, or adjacent. In other words, we know that New Writer is the opposition idea of *Commissie voor de Volkslectuur* [20], so in any discussion about Indonesia intellectuals, the both always contested, see Fig. 1.

5. CONCLUSIONS AND FUTURE WORK

The proposed similarity has the potential to be incorporated into enumerating for generating relations between objects. It shows how to uncover underlying strength relations by exploiting hit counts of search engine, but this work do not consider length of queries. Therefore, near future work is to further experiment the proposed similarity and look into the possibility of enhancing the performance of measurements in some cases.

References

[1] Grunwald, P. D., and Vitanyi, P. M. B. 2003. Kolmogorov complexity and information theory. *Journal of Logic, Language and Information* 12: 497-529.

[2] Sipser, M. 1996. *Introduction to the Theory of computations*, PWS Publishing C, Boston.

[3] Montaña, J. L., and Pardo, L. M. 1998. On Kolmogorov complexity in the real Turing machine setting. *Information Processing Letters*, 67: 81-86.

[4] Leung-Yan-Cheong, S. K., and Cover, T. M. 1978. Some equivalences between Shannon entropy and Kolmogorov Complexity. *IEEE Trans Info Theory*, Vol. IT-24, No. 3, May.

[5] Nannen, V. 2003. A short introduction to Kolmogorov Complexity. http://volker.nannen.com/work/mdl/

[6] D.R. Powell, Dowe, D. L., Allison, L., and Dix, T. I. Discovering simple DNA sequences by compression. Monash University Technical Report: monash.edu.au.

[7] Xiao, H. 2004. Komogorov complexity: computational complexity course report. Quenn's University, Kingnston, Ontario, Canada.

[8] Li, M. and Vitanyi, P. 1997. *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, NY.

[9] Romashchenko, A., Shen, A., and Vereshchagin, N. 2002. Combinatorial interpretation of Kolmogorov complexity. *Theoretical Computer Science*, 271: 111-123.

[10] Ziv, J., and Lempel, A. 1978. Compression of individual sequences via variable-rate encoding. *IEEE Trans Inform Theory* IT-24:530536

[11] Vitanyi, P. 2005. Universal similarity. In M. J. Dinneen et al. (eds.), *Proc. Of IEEE ISOC ITW2005 on Coding and Complexity*: 238-243.

[12] Cilibrasi, R. L., and Vitanyi, P. M. B. 2005. The Google similarity distance. *IEEE ITSOC Information Theory Workshop 2005 on Coding and Complexity*, 29th August-1st Sept. Rotorua, New Zealand.

[13] Bennett, C. H., Li, M., and Ma, B. 2003. Chain letters and evolutionary histories. *Scientific Am.*, June: 76-81.

[14] Burges, C. J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Vol. 2, No. 2: 121-167.

[15] Cilibrasi, R., Wolf, R. de, and Vitanyi, P. 2004. Algorithmic clustering of music based on string compression. *Computer Music J.*, Vol. 28, No. 4: 49-67.

[16] Cimiano, P. and Staab, S. 2004. Learning by Googling. *SIGKDD Explorations*, Vol. 6, No. 2: 24-33.

[17] Muir, H. 2003. Software to unzip identity of unknown compressors. *New Scientist*, Apr.

[18] Patch, K. 2003. Software sorts tunes. *Technology Research News*, 23, 30 Apr.

[19] Benett, C. H., Gács, Li M., Vitanyi, P. M. B., and Zurek, W. 1998. Information distance. *IEEE Transactions on Information Theory*, Vol. 4, No. 4, July: 1407-1423.

[20] Sutherland, H. 1968. Pudjangga Baru: Aspects of Indonesian Intellectual Life in the 1930s. *Indonesia*, Vol. 6, October: 106-127.

Mahyuddin K. M. Nasution: Departemen Matematika, FMIPA Universitas Sumatera Utara, Medan 20155, Indonesia
E-mail: mahyuddin@.ac.id